RHIC Ramps in the Feedback Age

Al Marusic

RHIC Retreat, July 26, 2012

This talk is about ramp preparation, ramp development, and general running issues.

What Was Done and What Should Be Done

This run required less effort than run 11. Partially that was due to the improvements made since last run. For this run, we made more changes than for many previous runs combined. The reason for that is that during run 11 we finally solved some long lasting problems with ramp development and that provided direction for further improvements.

Ramp Preparation

What was improved:

- The problem of specifying design orbits in an easy way had been solved (both in terms of specifying strengths of dipole correctors and the goal orbits for orbit feedback): the solution was making specification of design orbit part of the ramp (and specifying orbit using FeedbackEditor). Viewing of design orbits was also made simple. That saved hour(s) of ramp validation time.
- The procedure for fast generation of optics files for orbit feedbacks (and 10 Hz feedback) had been developed.
- Data presentation in RampEditor had been improved (separation of qmain / qtrim settings, better order of settings, accurate indication of anchoring of setpoints).

Remaining ramp related problems:

- Use of sds files, and related numbering of beam elements.
- cdev related problems.
- Smoothing of tunes / chromaticities.
- Not using splines to specify strengths of qgt power supplies.

- Certain ramp parameters can not be viewed / inspected by any application (RampDesigner should be modified to do so, made to enforce constraints).
- Simplify / reduce the number of changes needed when changing ramps. For example: different applications require that gammaT be specified as gammaT^2, 1/gammaT^2, etc.

Using Ramps Operationally

What was changed:

- Sequences involved in ramping were sped up.
- wfgman was sped up (from ~ 12 s to ~ 2 s for ramp).
- New way of handling the ramps was devised. As a result, it became possible to split RampManager & OptiCalc into operational and 9 other managers to hold ramps from previous years and ramps under development. Number of bugs which caused lots of disruption were fixed. New way of handling ramps made accessing ramp data from

scripts much easier (ORM, BBA, local coupling, ...).

- Bump manager had been created to allow scripts to make bumps (it is also used in conversion of design orbits to strengths, and also it calculates BPM responses).
- Orbit control was improved in several ways: ramping with zeroed correctors' strengths, converting correctors' strengths at injection from one ramp to another, improvement in correction algorithm (removing over-determination).
- Orbit feedback was used to make bumps (instead of RhicOrbitDisplay).

To do:

- Move all control of feedbacks (tune, orbit, replay) into wfgman.
- Fix various annoyances. For example there are number of problems with RhicInjection, such as not showing any data if some is missing, and issues with selection of fill patterns. Crucial programs have to be made more robust, and clearly report errors they encounter.
- Provide a way to view all beam elements.

Instrumentation

What was fixed:

- RhicChromaticity had been improved.

Things to fix:

- Better processing of WCM data.
- Retroactive processing of IPM data.
- Improve calculation of average orbit.
- Increase resolution of PS of dipole correctors from 12 to 16 bits.
- Develop procedures for measuring chromaticity using radius offset, modify existing algorithm to use DFT.
- Many ADOs have only generic names, give them system names. Provide descriptions for parameters.
- There are systems / programs which are not suitable for their roles: flagIo ADO has 41 parameters to manage 3 bits of information poorly.

Conclusions

It is time to start addressing “minor” issues, no more big time saving to achieve. Improving and speeding up things we did many times will leave more time for solving new problems. Complain more.